The Influence of Data Pre-processing and Post-processing on Long Document Summarization

Kailun Dong, Xinwei Du, Yuchen Zhang, Yongsheng Li, Ruei-Yu Tsay

Abstract

Long document summarization is an important and hard task in the field of natural language processing. A good performance of the long document summarization reveals the model has a decent understanding of the human language. Currently, most researches focus on how to modify the attention mechanism of the transformer to achieve a higher ROUGE score. The study of data pre-processing and post-processing are relatively few. In this paper, we use two pre-processing methods and a post-processing method and analyze the effect of these methods on various long document summarization models. [1]

1 Introduction

Long document summarization is a hard and important task which requires the model to identify and extract the important information in the document and generate a fluent summary. A good performance in the long document summarization usually shows the model's decent understanding of natural language.

Long document summarization is a sequence to sequence task, and a general way to handle this task is by using an LSTM encoder and decoder with the attention mechanism. Discourse-Aware (Cohan et al., 2018) is a pioneering paper in the long document summarization, they provided two standard datasets (arXiv and pubMed) and proposed a hierarchical encoder and a discourse-aware decoder to generate the summary. They also adopted copying mechanism to address the problem of unknown tokens and used a decoder coverage vector to avoid repeated phrases in the summary. TLM model (Subramanian et al., 2019) combined both extractive and abstractive methods and used a transformer model (Vaswani et al., 2017) to generate the summary. T5 (Raffel et al., 2019) is a language model which explored the transfer learning techniques and introduced a unified text-to-text framework. BigBird (Zaheer et al., 2021) proposed a sparse attention mechanism and reduced the attention complexity from quadratic to linear. Longformer and its variant LED (Beltagy et al., 2020) further modified the attention mechanism and combined local windowed attention with global attention. HEPOS (Huang et al., 2021) proposed a novel efficient encoder-decoder attention and achieved the state-of-the-art results on PubMed dataset.

Nowadays, more and more long document summarization researches focus on how to improve the attention mechanism of the transformer. However, other research areas, like image retrieval, for example, have post-processing methods like PQ (Jégou et al., 2011), DBA (Philbin et al., 2007), and QE (Chum et al., 2011). These post-processing methods can further improve the performance of the model and the implementation of the above-mentioned post-processing methods are independent of the core image retrieval algorithms. Whereas, decent pre-processing and post-processing methods are lacking in the area of long document summarization.

In this paper, we use two pre-processing methods and a post-processing method and analyze the effect of these methods.

2 Methods

Figure 1 shows our long document summarization pipeline. For a given document, we first use a pre-processing method to extract important information from the original document. Later, we feed the extracted text to a long document summarization model and generate the summary. Then, we

[1] You can find our implementation on Github: `https://github.com/Anthonyive/csci-544-project`. Our video demo:`https://www.youtube.com/watch?v=oVIVtOPeWEs`

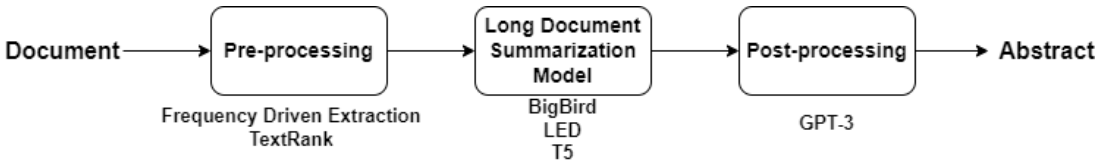

Figure 1: Long Document Summarization Pipeline.

use a post-processing method to refine the summary generated by the model and form our final abstract.

2.1 Pre-processing

We use extraction-based methods to fulfill data pre-processing. The extraction-based method is using some ranking algorithms to extract important sentences from original documents. Here in this paper, we used two different approaches of extractive methods to tackle the summarization problem.

2.1.1 Frequency Driven Extraction

We first tried a Frequency Driven Approach. This method is useful for several reasons: **(1)** It gives us a baseline of what the target rouge-1, rouge-2, rouge-L scores are going to be; **(2)** It could become the first processing step for later steps; **(3)** Its fast, reliable, and light processing step makes it easy to pipe to other methods.

Here we used a very simple scoring algorithm: Given a document D with N sentences, $D = \{S_1, S_2, ..., S_N\}$, we first remove stopwords and assign each remained word w_i with equal weight 1 (we can also give values other than 1 based on prior knowledge, we set all weight to 1 for simplicity). Then, we calculate the value of each word v_{w_i} by calculating the total weight of word w_i in this document (in this case, we try to calculate the frequency of each word in this document since we set all weight to 1).

$$v_{w_i} = count(D, w_i) \tag{1}$$

For each sentence S_i, the value of each sentence V_{S_i} euqals to the sum of the value of the words in the sentence.

$$V_{S_i} = \sum_{w_i \in S_i} V_{w_i} \tag{2}$$

Finally, we choose the top M sentences to create the summary. M is a hyperparameter. And we set M as the nnumber of sentences of the ground true summary.

$$Result = top\_M(D) \tag{3}$$

We acknowledge that the top_M approach might be limited as M is dependent on the reference summary data. However, we are interested in finding the differences between whether using this frequency driven approach will increase the train and dev data accuracies or not.

Figure 2 show the flowchart of frequency driven algorithm.

2.1.2 TextRank Algorithm

The choosing of M could be a problem for the algorithm in section 2.1.1 when we try to summary new data. To resolve finding M, we choose two approaches: One is we treat this M as a fixed hyperparameter, i.e. a fixed number of sentences to find in the predicted summary. Another one is to use another approach without using M.

We use a graph-based method, TextRank (Mihalcea and Tarau, 2004), which was influenced by Google's Page Rank Algorithm (Page et al., 1999), to do extractive document summarization.

2.2 Long Document Summarization Models

We adopt various long document summarization model to perform the summarization. We use fine-tuned BigBird and LED models, and trained a T5 model from scratch.

A benefit of our summarization pipeline is that the implementation of long document summarization won't affect the data pre-processing step and post-processing step.

2.3 Post-processing

We use GPT-3 model to handle the post-processing step. GPT-3 model is a transformer-based language model and is trained on large datasets which contain a huge amount of information about the reading text for many kinds of areas. Lots of experiments show that GPT-3 has a strong language generating ability. Therefore, in our experiment, we use the GPT-3 model to do the fine post-processing step.

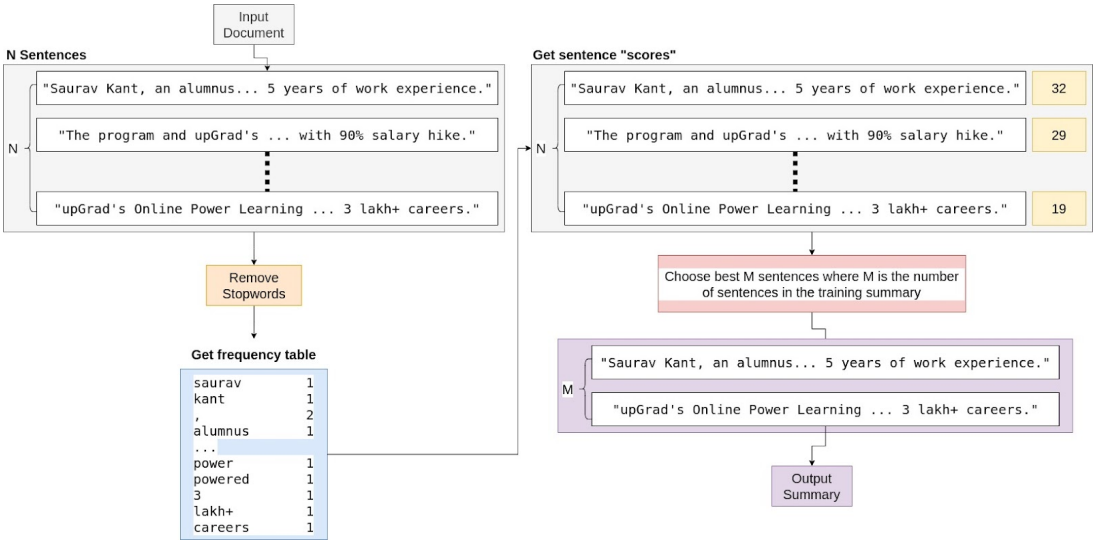

Figure 2: The Flowchart of Frequency Driven Algorithm.

Given the summary proposed by a long document summarization model, we use prompt as *"Original [summary], Polished Sentence:"*. GPT-3 model will complete this prompt and generate the polished sentences. We adopt the curie engine of GPT-3 which has a decent computing speed and performance.

3 Experiments and Results

3.1 Datasets

We used two datasets, namely arXiv and PubMed, to train our models and evaluate our models' performances.

The arXiv dataset and the PubMed dataset are both free and open-access resources for research use. Because the original full-size datasets are too large, the arXiv and PubMed datasets we used are subsets of their original data that take storage space over 1T and 90G. The arXiv dataset and PubMed dataset consist of 215913 and 133215 published scholarly articles respectively, each having been separated into a training set, a validation set, and a testing set with similar distributions on articles' sentence counts and token counts. The datasets also contain section information and citation information of the articles, while our team mainly focuses on the article texts to extract the articles' summarizations.

Table 1 shows the distribution characteristics of the two datasets.

3.2 Model Setting

In order to better test the generalization ability of our proposed method, we use a variety of long document models. The following describes how we use these models.

BigBird: We used two BigBird (Zaheer et al., 2021) models fintuned on arXiv and PubMed datasets respectly to obtain the Rouge performance of BigBird model on these two datasets. The maximum number of input tokens for BigBird is 4096. To generate the summary, we set the length penalty as 0.8, number of beam search as 5 and the maximum number of output tokens as 256.

LED: We used Longformer-Encoder-Decoder (LED), a variant of Longformer model (Beltagy et al., 2020), designed for long document summarization. We used two fintuned LED models pretrained on arXiv and PubMed datasets respectly. The maximum number of input tokens of LED is 16384, but because of the limitations of GPU memory, we set the maximum number of input tokens as 8192. Longformer proposed a global attention mechanism, which can help the model to understand the documents better. We put gloabl attention on the start token of each sentence.

T5: T5 (Raffel et al., 2019) is a unified Text-to-Text Transfer Transformer pre-trained on a large text corpus and has been demostrated to achieve the state-of-the-art performance on many NLP

Table 1: Dataset Distribution Characteristics.

Dataset	Num of doc	Avg num of sen	Avg num of tokens (doc)	Avg num of tokens (sen)
arXiv	215k	206	6029	29
PubMed	133k	86	3048	33

Table 2: Summarization ROUGE score for long documents.

	arXiv			PubMed		
Model	R-1	R-2	R-L	R-1	R-2	R-L
BigBird	**41.94**	**21.59**	**36.56**	**42.45**	18.18	**37.74**
+ Extraction	27.54	7.63	17.06	34.21	10.89	20.65
+ TextRank	35.58	10.36	19.59	42.15	**27.80**	32.95
+ GPT-3	42.16	21.86	36.75	34.95	12.73	31.07
LED	**45.96**	**26.97**	**42.48**	43.10	17.11	**32.76**
+ Extraction	37.13	12.46	28.01	41.56	18.95	24.02
+ TextRank	40.43	11.23	25.00	**49.53**	**21.45**	29.48
+ GPT-3	38.22	21.46	35.67	38.39	14.40	27.68
T5 (Fail)	12.65	5.45	10.82	14.42	5.58	12.99
Extraction (Only)	27.05	7.94	13.87	34.65	12.40	19.66
TextRank (Only)	29.49	9.58	14.86	31.34	12.25	17.35

tasks. We fine-tuned long document summarization on the T5 pre-trained model using PubMed and Arxiv dataset, the maximum input length is set to 4096 and the maximum generation length is set to 256 due to GPU memory limits.

3.3 Results and Discussion

Table 2 shows the Rouge score of the BigBird model and LED model, and the score when we apply pre-processing and post-processing methods. (Due to the limitation of computing resources and time, We only tested the first 100 test data in each dataset. Therefore, instead of focusing on the specific Rouge value, please focus on the difference in Rouge between different methods.) We also listed the performance of our T5 model, which is trained from scratch, and the performance when we only use the pre-processing methods.

According to Table 2, pre-processing method TextRank may increase the performance sometimes and decreases the performance at other time. Pre-processing method finds the important sentence. It will help long document summarization models discard unimportant content and reduce distractions. However, the pre-processing methods are not useful all the time, when it extracts unimportant content and discard important sentence by mistake, this method will decrease the summary performance.

As for post-processing, when we try to apply GPT-3 as a post-processing method, the Rouge score of the model decreases. It makes sense since we only give GPT-3 model the summary generated by the long document summarization model. We try to input the original whole paper but it is too long to be handled by GPT-3.

Moreover, T5 does not perform well on long document summarizations according to the Rouge score, though it claimed to get good performance in summarization in the original T5 paper on short documents of CNN news. We think this may be because that the average length of CNN news is only 656 will the average length of arXiv and Pubmed datasets are more than 3000. In addition, the abstractive summary generated by the T5 model is very short, the original summary of PubMed has a median length of 226 and only has a median length of 10 for generated summary.

4 Conclusions and Future Work

To explore how to further improve the performance of the long document summarization, we used frequency driven extracted algorithm and TextRank algorithm for data pre-processing, and use GPT-3 for post-processing. The experiment results show that our proposed method can not improve the performance of the model.

In future work, we need to why study will T5 model generate such short summaries and if there are ways to modify it to adapt to long document summarization. We also want to try larger models such as T5-base and T5-large to see if we can improve the performance.

Moreover, we believe a decent post-processing method is needed not only for long document summarization, but also for other language generation tasks such as machine translation, dialogue generation. A model dedicated to polishing articles may be useful for NLP tasks.

References

[Beltagy et al.2020] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[Chum et al.2011] Ondřej Chum, Andrej Mikulík, Michal Perdoch, and Jiří Matas. 2011. Total recall ii: Query expansion revisited. In *CVPR 2011*, pages 889–896.

[Cohan et al.2018] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

[Huang et al.2021] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*.

[Jégou et al.2011] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. 2011. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716.

[Mihalcea and Tarau2004] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

[Page et al.1999] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

[Philbin et al.2007] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE.

[Raffel et al.2019] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

[Subramanian et al.2019] Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. 2019. On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186*.

[Vaswani et al.2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

[Zaheer et al.2021] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.